PATENT ABSTRACTS OF JAPAN

(11)Publication number : 58-225469
(43)Date of publication of application : 27.12.1983

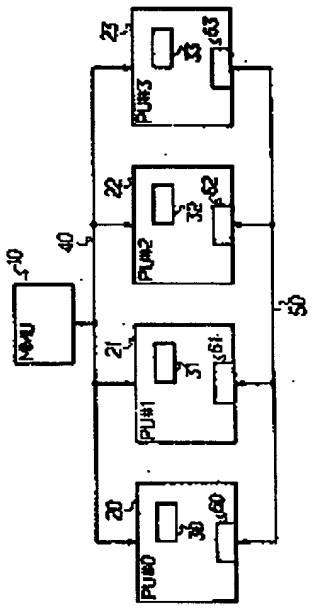
(51)Int.Cl. G06F 15/16
G06F 13/00

(21)Application number : 57-109635
(22)Date of filing : 25.06.1982

(71)Applicant : NEC CORP
(72)Inventor : TAKAGI HAJIME

(54) MULTI-PROCESSOR CONTROLLING SYSTEM

(57)Abstract:
PURPOSE: To minimize the change of a monitor program, by selecting one of idle processor units and indicating the execution of an instruction following a program state word loading instruction.
CONSTITUTION: A multi-processor controlling system comprises a memory unit 10, plural processor units 20W23, a memory bus 40 and a control bus 50. The memory unit 10 stores plural programs. The processors 20W23 share the unit 10 and execute simultaneously at least one of plural programs. When one of processors 20W23 starts execution of a new program by a program state word loading instruction, one of other idle processor units is selected. Then the execution is indicated to this selected processor unit for an instruction following the program state word loading instruction.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑲ 日本国特許庁（JP）　　⑪ 特許出願公開

⑫ 公開特許公報（A）　　昭58—225469

⑤Int. Cl.³	識別記号	庁内整理番号
G 06 F 15/16		Z 6619—5B
13/00	101	7218—5B

⑬公開　昭和58年(1983)12月27日

発明の数　1
審査請求　未請求

（全 7 頁）

⑤④マルチプロセッサ制御方式

⑳特　　願　昭57—109635
㉒出　　願　昭57(1982)6月25日
㉑発 明 者　高木一
東京都港区芝五丁目33番1号日本電気株式会社内
⑦①出 願 人　日本電気株式会社
東京都港区芝5丁目33番1号
⑦④代 理 人　弁理士　井ノ口壽

明　細　書

1 発明の名称

マルチプロセッサ制御方式

2 特許請求の範囲

複数のプログラムを記憶しておくためのメモリユニットと、前記メモリユニットを共用し、前記複数のプログラムのうちの少なくともひとつのプログラムを同時に実行するための複数個のプロセッサユニットと、前記メモリユニットと前記複数個のプロセッサユニットとを接続するためのメモリ母線と、前記複数個のプロセッサユニットの相互の間を接続するための制御母線とを具備し、且つ、前記複数個のプロセッサユニットのそれぞれがインストラクションカウントレジスタと前記制御母線に接続されたプロセッサ間通信制御部とを具備し、前記複数個のプロセッサユニットのうちのひとつがプログラム状態語ロード命令により新たなプログラムの実行を開始するときに他のプロセッサユニットのなかでアイドル状態となつているものがある場合には、前記アイドル状態となつているプロセッサユニットのうちのひとつを選択し、前記プログラム状態語ロード命令に続く命令の実行を前記選択されたプロセッサユニットに対して指示することを特徴としたマルチプロセッサ制御方式。

3 発明の詳細な説明

（発明の属する技術分野の説明）

本発明は電子計算機システムにおけるマルチプロセッサ制御方式、特に複数個のプロセッサユニットで共用されたメモリユニットを含むマルチプロセッサシステムにおいてプログラムの割付けを行うためのマルチプロセッサ制御方式に関する。

（従来技術の説明）

従来からマルチプロセッサシステムにおいては、複数個のプロセッサユニットのそれぞれに対して実行すべきジョブプログラムを割付けることが必要であり、この割付けの制御はモニタ

プログラムと呼ばれる特殊な制御プログラムにより行われることが公知である。モニタプログラムは実行可能なジョブプログラムを選択してプロセッサユニットに割付ける機能を有するもので、マルチプロセッサシステムのモニタプログラムはシステム内に存在するプロセッサユニットの数量、および各プロセッサユニットの状態などを管理する必要がある。しかしながら、単一のプロセッサユニットのみを有するシングルプロセッサシステムにおいてモニタプログラムを実行する場合には複数箇のプロセッサユニットを意識する必要がないので、これをそのままマルチプロセッサシステムに適用しようとすると、モニタプログラム自体の大幅な変更が必要となる。しかし、シングルプロセッサシステムからマルチプロセッサシステムへの移行は電子計算機システムの成長の課程として多くの場合必要となるものである。このような場合、モニタプログラムの変更は一般にその制御下で動作するジョブプログラムの変更を伴う場合が多く、当業者にとつてはその変更量をいかにして最小限にとどめるかが大きな課題となつている。

（発明の目的の説明）

本発明の目的は、シングルプロセッサシステムで使用されるモニタプログラムに最小限の変更を行うだけでマルチプロセッサシステムのためのモニタプログラムとして適用し得るように構成したマルチプロセッサ制御方式を提供することにある。

（発明の構成の説明）

本発明によるマルチプロセッサ制御方式はメモリユニットと、複数箇のプロセッサユニットと、メモリ母線と、制御母線とを具備して実現したものである。メモリユニットは複数のプログラムを記憶するためのものである。複数箇のプロセッサユニットはメモリユニットを共用し、複数のプログラムのうち、少なくともひとつのプログラムを同時に実行するためのものである。各プロセッサユニットはインストラクションカウントレジスタとプロセッサ間通信制御部とを具備したものである。メモリ母線はメモリユニットと複数箇のプロセッサユニットとを接続するものであり、情報がこのメモリ母線を通つて転送される。制御母線は複数箇のプロセッサユニットの相互の間を接続するもので、制御母線は各プロセッサユニットの内部のプロセッサ間通信制御部に接続されている。

複数箇のプロセッサユニットのうちのひとつがプログラム状態語ロード命令によつて新たにプログラムの実行を開始するときに他のプロセッサユニットのなかでアイドル状態となつているものがある場合には、アイドル状態にあるプロセッサユニットのうちのひとつを選択し、プログラム状態語ロード命令に続く命令を実行するように、選択されたプロセッサユニットに対して指示をする。

（発明の原理と作用の説明）

本発明の一実施例を説明する前に、第1図に示した様な単一のプロセッサユニットのみで構成したシングルプロセッサシステムにおいて、モニタプログラムがプロセッサユニットにジョブプログラムを割付ける方法を説明する。

第1図を参照すると、シングルプロセッサシステムはメモリユニット1とインストラクションカウントレジスタ3とを具備したプロセッサユニット2、ならびにメモリユニット1とプロセッサユニット2との間を接続するための信号線4から構成されている。プロセッサユニット2は、プログラムを実行するユニットであり、プログラムを構成する各命令を逐次、インストラクションカウントレジスタ3の内容に従つてメモリユニット1から取り出して実行する。

第2図を参照すると、本発明に依るマルチプロセッサ制御方式によつて定義されているプログラム状態語（ＰＳＷ）はフィールド1、フィールド2およびフィールド3に分割されている。このプログラム状態語はプログラムを実行するために必要な情報を有し、各プログラム毎にその情報は固有なものである。

上記フィールド1の内容は該当するプログラム

状態語に属するプログラムの実行モードを決定するものであり、フィールド2は該当するプログラムの最初の命令のメモリユニット1の内部での位置、すなわちアドレス情報を指示するものである。これは、特にインストラクションカウンタと呼ばれる。上記フィールド3にはプログラム実行中における入出力割込みに関するマスク情報を備えている。既に説明した様に、プログラム状態語は各プログラム毎に定義されているものであるため、プログラムの個数だけのプログラム状態語が存在することになり、メモリユニット1の内部にプログラムとともに格納されている。第3図はメモリユニット1の内部において各プログラムとプログラム状態語とが結びついて格納されている状態を概念的に示したものである。第3図を参照すると、それぞれモニタプログラム、ジョブプログラムa、ジョブプログラムb、およびジョブプログラムcに固有なプログラム状態語がそれぞれPSW0、PSWa、PSWb、およびPSWcとして割当られている。また、プログラム状態語PSW0、PSWa、PSWb、およびPSWcのメモリ内部における格納アドレスはそれぞれ1000_{16}、2000_{16}、2010_{16}、および2020_{16}である。ここで、添字の16は16進数表示であることを示す。また、モニタプログラム、ジョブプログラムa、ジョブプログラムbおよびジョブプログラムcのメモリユニット1の内部での開始アドレス、すなわち各プログラムの中で最初に実行すべき命令のアドレスはそれぞれ5000_{16}、6000_{16}、7000_{16}、および8000_{16}であるとする。従つて、図示された各プログラム状態語のフィールド2、すなわちインストラクションカウント部の内容は、それぞれ5000_{16}、6000_{16}、7000_{16}、および8000_{16}となつている。

以上説明したように、プログラムの実行においては、プログラム状態語により該当プログラムの動作態様、およびメモリユニット1の内部でのプログラムの格納アドレスを認識することができる。したがつて、これらのプログラムをプロセッサユニット2により実行するためには、実行を開始しようとするプログラムに対応するプログラム状態語をプロセッサユニット2に取込み、プロセッサユニット2の内部のプログラム実行制御部に与えればよい。このために、特にプログラム状態語ロード命令（LOAD PSW命令）が用意されている。

第4図は、プログラム状態語ロード命令の構成を図示したものである。この命令はOPコードフィールドに983_{16}を有する命令として定義され、アドレスフィールドにはプログラム状態が格納されているメモリアドレスを含んでいる。プログラム状態語ロード命令は「アドレスフィールドにより指示されているメモリアドレスからプログラム状態語を取出し、プロセッサユニット2の内部の所定のレジスタにロードする」という機能を有している。この所定のレジスタとはプログラム状態語の各フィールドの内容を保持するためのものであり、特に第2図に示すフィールド2の内容、すなわちインストラクションカウント部の内容は第1図に示すインストラクションカウントレジスタ3に格納される。すなわち、プログラム状態語ロード命令によりプロセッサユニット2の内部のインストラクションカウンタレジスタ3はそれまでとは別の新しい情報であるインストラクションカウント部を含むものである。インストラクションカウントレジスタ3にセットされている以外の他のプログラム動作モードについても同様に実行されるものである。プロセッサユニット2はインストラクションカウントレジスタ3の内容に応じ、メモリユニット1の内部から命令を取出して実行することによつてプログラムを実行していく。この点は既に説明したとおりであり、このプログラム状態語ロード命令を実行した後はプロセッサユニット2はそれまでとは別のプログラムを実行していくことになる。換言すれば、プロセッサユニット2の内部で実行するプログラムの切換えを行つたことになる。以上の機能を使用し、モニタプログラムはジョブプログラムを

プロセッサユニット2へ割付ける。すなわち、例えば第3図のように実行開始を待つているジョブプログラムa，b，cが存在する場合、モニタプログラムは前もつて決められている選択基準に従つてそのうちのひとつ、例えばジョブプログラムaを選択し、対応するプログラム状態語をプロセッサユニット2に与える。しかしながら、モニタプログラム自身もひとつのプログラムであるので、現在、プロセッサユニット2の内部で実行中のプログラムはモニタプログラムである。すなわち、プロセッサユニット2の保有するプログラム状態語はモニタプログラム用のプログラム状態ＰＳＷ0である。従つてプロセッサユニット2の実行プログラム、すなわち換言すればプログラム状態語（ＰＳＷ）をモニタプログラムからジョブプログラムaに切り換える必要がある。このため、上記プログラム状態語ロード命令によつてジョブプログラムaに対応するプログラム状態語Ｐｓｗａがプロセッサユニット2にロードされる。

第5図はモニタプログラムがジョブプログラムをプロセッサユニット2に割付ける場合のフローチャートを示す図である。第5図に示すように、まず、実行可能なジョブプログラムの中から選択的にひとつのプログラムを決定し、このプログラムを実行状態として区別した後、対応するプログラム状態語をプログラム状態語ロード命令によりプロセッサユニット2にロードする。第5図に示すフローチャートにおいて、特に重要な点はプログラム状態語ロード命令を実行した後では処理が行われない点である。なぜならば、このモニタプログラムはシングルプロセッサシステムのためのモニタプログラムであり、プログラムを実行するプロセッサユニット2はひとつしか存在しないためである。このため、ジョブプログラムがプロセッサユニット2に割付けられた後は、もはや他にプログラムの実行できるプロセッサユニットは存在しない。従つて、モニタプログラムはジョブプログラムをプロセッサユニットに割付けた後、自身の動作するプロセッサユニットがないのでその後の処理は無意味となる。以上の説明からシングルプロセッサシステムの場合にモニタプログラムのみがジョブプログラムをプロセッサユニット2に割付けることは明らかである。

（実施例の説明）

次に、この原理を応用して構成した本発明に依るマルチプロセッサ制御方式の一実施例について、図面を参照して説明する。

第6図は本発明に依るマルチプロセッサ制御方式の一実施例を示す。第6図において、マルチプロセッサ制御方式はメモリユニット10と、第1～第4のプロセッサユニット20～23とから構成されている。第1～第4のプロセッサユニット20～23は共通のメモリ母線40によりメモリユニット10に結合されている。また、第1～第4のプロセッサユニット20～23の内部には、それぞれ互いに他のプロセッサユニットとの情報の授受を行うための第1～第4のプロセッサ間通信制御部60～63が具備されている。それぞれ、第1のプロセッサユニット20には第1のプロセッサ間通信制御部60、第2のプロセッサユニット21には第2のプロセッサ間通信制御部61、第3のプロセッサユニット22には第3のプロセッサ間通信制御部62、第4のプロセッサユニット23には第4のプロセッサ間通信制御部63が対応する。第1～第4のプロセッサ間通信制御部60～63は共通の制御母線50により結合されている。第1～第4のプロセッサユニット20～23は第1図に示すシングルプロセッサシステムと同様なプログラム実行ユニットであり、それぞれ第1～第4のインストラクションカウントレジスタ30～33を具備する。

次に、第6図に示したマルチプロセッサ制御方式における一実施例のモニタプログラムの動作を第7図のフローチャートに従つて説明する。第7図を参照すると、ステップＡ，Ｂ，Ｃはシングルプロセッサシステムの場合と同様な動作をするが、ステップＤのブランチ命令の実行の後でステップＡに戻るように構成されている点

が第3図のシングルプロセッサシステムの場合とは異なる。すなわち、ステップA→ステップB→ステップCの移行により1個のジョブプログラムが第1～第4のプロセッサユニット20～23のひとつに割付けられた後、さらにステップDのブランチ命令によりステップAに戻る。以下、同様の処理が繰返される。再び第6図を参照すると、第1～第4のプロセッサユニット20～23では、それぞれプログラム状態語ロード命令によつて、自身のプログラム状態語から該当する命令によつて指定されたプログラム状態語へと切換えられるが、この動作はシングルプロセッサの場合と同様である。しかし、本発明に依る上記実施例においてこの切換え動作を行う場合には、第1～第4のプロセッサユニット20～23の内部にそれぞれ対応して存在する第1～第4のプロセッサ間通信制御部60～63と、共通の制御母線50とを介し、他のアイドル状態にあるプロセッサユニットのうちのひとつを選択する。そこで、それまで保有していたプログラム状態語を選択されたプロセッサユニットに対して転送する。すなわち、モニタプログラムを実行中であつたプロセッサユニットはそれまで実行中であつたモニタプログラムのプログラム状態語をアイドル状態にある他のプロセッサユニットのひとつに与えた後、プログラム状態語ロード命令で指定された新しいジョブプログラムのプログラム状態語を自身にロードし、このジョブプログラムの実行を開始する。ここで、モニタプログラムを実行するプロセッサユニットを第1のプロセッサユニット20とし、アイドル状態にあつて第1のプロセッサユニット20により選択されたプロセッサユニットを第2のプロセッサユニット21とすると、第1のプロセッサユニット20はモニタプログラムのプログラム状態語を第2のプロセッサユニット21に与えた後、プログラム状態語ロード命令で指定された新しいジョブプログラムのプログラム状態語を第1のプロセッサユニット20にロードする。これによつて、第1のプロセッサユニット20はジョブプログラムの実行を開始する。いつぽう、第2のプロセッサユニット21は第1のプロセッサユニット20の指示により第1および第2のプロセッサ間通信制御部60，61と制御母線50とを介し、モニタプログラムのプログラム状態語をロードするように指示を受けると、与えられたモニタプログラムのプログラム状態語に従つてプログラムの実行を開始する。このとき、プログラム状態語は第1のプロセッサユニット20がプログラム状態語ロード命令を実行した後の状態であるので、プログラム状態語のインストラクションカウント部は第7図のプログラム状態語ロード命令の直後に位置するステップDのブランチ命令を指示している。依つて、第2のプロセッサユニット21は第7図のステップDのブランチ命令を実行するので、ステップAに戻つてステップA→ステップB→ステップCの処理が順次続行される。従つて第1のプロセッサユニット20で以前実行したのと同様にして第2のプロセッサユニット21でモニタプログラムを実行し、第7図のステップCで示されたプログラム状態語ロード命令に到達すると、再び第2のプロセッサユニット21はアイドル状態である他のプロセッサユニットを選択し、モニタプログラムのプログラム状態語をこのプロセッサユニットに与えた後、ジョブプログラムの実行を開始する。このようにして次々とモニタプログラムがアイドル状態になつているプロセッサユニットに移され、そのプロセッサユニットによつてジョブプログラムが起動される。

第8図は第1のプロセッサユニット20から始まり、第3のプロセッサユニット22までジョブプログラムの起動が行われる様子を時系列的に示した図である。第8図ではジョブプログラムa，b，cが実行開始を待つており、この順に各ジョブプログラムが選択される。第1のプロセッサユニット20がモニタプログラムからジョブプログラムaに切換ると、第2のプロセッサユニットでモニタプログラムの実行を開始する。第2のプロセッサユニット21でモニタプログラ

ムからジョブプログラムbに切換わると、第3のプロセッサユニット22でモニタプログラムの実行を開始する。第3図のプロセッサユニット22でモニタプログラムからジョブプログラムcに切換わると、第4のプロセッサユニット23でモニタプログラムの実行を開始する。本実施例では、第4のプロセッサユニット23はアイドル状態にあるが、起動すべきジョブプログラムは存在しないので、モニタプログラムは再び第4のプロセッサユニット23をアイドル状態にさせておく。

4.図面の簡単な説明

第1図はシングルプロセッサシステムの実例のブロック図、第2図は本発明に依るマルチプロセッサ制御方式によつて定義されているプログラム状態語の構成を示す図、第3図はメモリユニットの内部において各プログラムとプログラム状態語とが結びついて格納されている状態を概念的に示したブロック図、第4図はプログラム状態語ロード命令の構成を図示した図、第5図はモニタプログラムがジョブプログラムをプロセッサユニットに割付ける場合のフローチャートを示す図、第6図は本発明によるマルチプロセッサ制御方式の一実施例を示すブロック図、第7図は第6図のマルチプロセッサ制御方式において、モニタプログラムがジョブプログラムをプロセッサユニットに割付ける場合のフローチャート、第8図は第1図のプロセッサユニットから始まり第3のプロセッサユニットまでのジョブプログラムの起動が行われる様子を時系列的に示した図である。

1，10…メモリユニット

2，20～23…プロセッサユニット

3，30～33…インストラクションカウントレジスタ

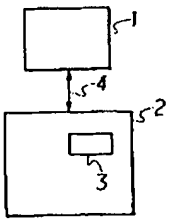
60～63…プロセッサ間通信制御部

40…メモリ母線　　50…制御母線

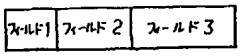
特許出願人　日本電気株式会社

代理人　弁理士　井ノ口　壽

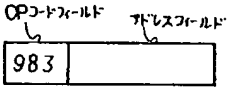
第1図



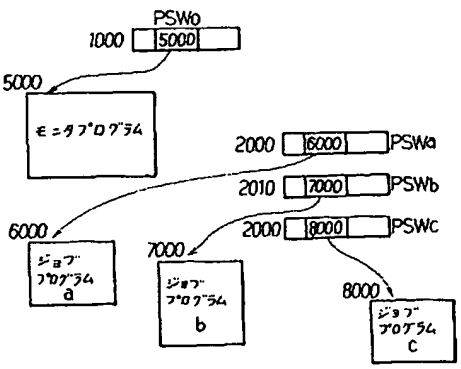
第2図



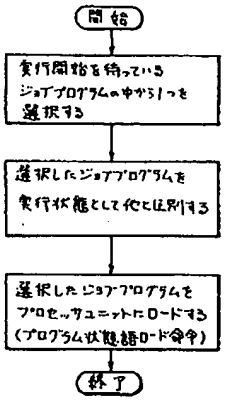
第4図



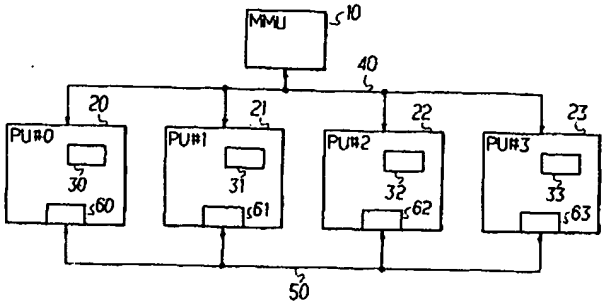
第3図



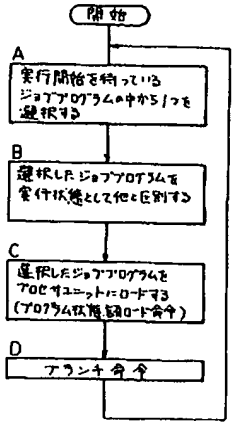
第5図



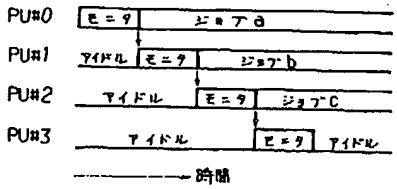
第6図


MMU
10
40
20
21
22
23
PU#0
PU#1
PU#2
PU#3
30
31
32
33
60
61
62
63
50

第7図


開始
A
実行開始を待っている
ジョブプログラムの中から1つを
選択する
B
選択したジョブプログラムを
実行状態として他と区別する
C
選択したジョブプログラムを
プロセサユニットにロードする
(プログラム状態語ロード命令)
D
ブランチ命令

第8図


PU#0
PU#1
PU#2
PU#3
モニタ
ジョブa
アイドル
モニタ
ジョブb
アイドル
モニタ
ジョブc
アイドル
モニタ
アイドル
時間